

AMENDMENTS

IN THE CLAIMS:

Please amend claims 1-3, 6-9, 12, 14, 16, 17, 20, and 22 and cancel claim 24 as provided below:

1. (Currently Amended) A layered network device driver system, operably stored and executable on a host computer system having an operating system, storage means and a processor, the layered network device driver system further comprising:

an operating system dependent module configured to translate commands from the operating system into a format independent of the operating system, operable to communicate in an operating system dependent format to initiate sending of transmit data in an operating system dependent format and to provide received data in an operating system dependent format;

an operating system independent module [[that]] configured to communicate[[s]] with the operating system dependent module in an operating system independent format and perform[[s]] processing on requests from the operating system dependent module and error checking on the transmit data and the received data in a manner that is independent of the operating system;

a [[media]] network device independent module configured to communicate specific network device information to the operating system independent module, wherein the specific network device information is communicated in a network device format that is independent of a network device associated with the host computer system, [[that]] place[[s]] the transmit data in one or more transmit descriptor rings, [[and]] obtain[[s]] the received data from one or more receive descriptor rings that reside in the host computer system, and selectively append the transmit or receive data with optional parameters; and

a [[media]] network device dependent module configured to communicate with the [[media]] network device independent module and with one or more types of the network device[[s]], and controlling the one or more types of network device[[s]] to send

the transmit data from the one or more transmit descriptor rings, ~~and to receiving~~ the received data into the one or more receive descriptor rings, and provide information to the network device dependent module that facilitates the sending and transmitting of data.

2. (Currently Amended) The system of claim 1, wherein the ~~[[media]]~~ network device dependent module is operable to configure and initialize one or more registers of ~~[[a]]~~ the network device.

3. (Currently Amended) The system of claim 1, wherein the ~~[[media]]~~ network device dependent module is operable to verify that the optional parameters comply with capabilities of ~~[[a]]~~ the network device.

4. (Original) The system of claim 1, wherein the operating system dependent module is operable to receive a packet to be transmitted from the operating system in an operating system specific format.

5. (Original) The system of claim 4, wherein the operating system dependent module is further operable to convert the packet into an array based data structure, wherein the data structure comprises virtual pointers to one or more data buffers.

6. (Currently Amended) The system of claim 5, wherein the operating system independent module is operable to convert the virtual pointers of the array based data structure into physical pointers, and
operable to communicate with a plurality of operating system dependent modules associated with a respective operating system.

7. (Currently Amended) The system of claim 6, wherein the ~~[[media]]~~ network device independent module is operable to attach the one or more data buffers to a

transmit descriptor, wherein the transmit descriptor is of the one or more transmit descriptor rings.

8. (Currently Amended) The system of claim 1, wherein the [[media]] network device independent module is operable to access [[on]] one or more data buffers of a packet associated with a received frame along with optional information, and operable to communicate with a plurality of network device dependent modules associated with a respective network device.

9. (Currently Amended) The system of claim 8, wherein the [[media]] network device dependent module is operable to analyze the packet and optional information for status, and determine if the packet is less than a predetermined size limit with respect to a coalescing size, and if so, coalesce the packet and buffers with one or more other packets and buffers.

10. (Original) The system of claim 9, wherein the operating system dependent module analyzes the packet for errors.

11. (Original) The system of claim 10, wherein the operating system dependent module is further operable to provide the packet to the operating system.

12. (Currently Amended) The system of claim 1, further comprising a message block format for transferring packets between the operating system and the operating system dependent module for receive operations, the message block comprising:

a mandatory parameter that includes one or more virtual pointers to one or more data buffers that contain data for a received frame; and
an optional parameter pointer that points to one or more optional parameters.

13. (Original) The system of claim 12, wherein the one or more optional parameters respectively include a type, a size, and one or more parameters.

14. (Currently Amended) A method of processing a received frame comprising:
obtaining a received packet with a network device and storing the received packet into a data buffer and generating an array based data structure from the received packet that includes one or more virtual pointers to one or more data buffers;
appending the array based data structure, by filling in a null pointer with an optional array that consists of optional parameters and size, by a network device independent module;
performing error checking on the appended array by an operating system independent module;
converting each virtual pointer to a physical address by a network device independent module by an operating system dependent module;
analyzing status information of the received packet and duplicating the received packet;
causing the one or more data buffers associated with the received packet to be freed;
analyzing the duplicated packet for errors; and
passing the duplicated packet to an operating system for further processing.

15. (Original) The method of claim 14, further comprising attaching the one or more freed data buffers to an adapter queue for incoming received frames.

16. (Currently Amended) The method of claim 14, wherein the received packet [[is]] received by [[a]] the network device [[and]] is placed initially in an adapter queue.

17. (Currently Amended) The method of claim 16, wherein ~~a media~~ the network device independent layer directly removes the received packet from the adapter queue.

18. (Original) The method of claim 14, wherein duplicating the packet comprises placing a duplicate copy of the packet in a duplicate queue.
19. (Original) The method of claim 14, wherein [[an]] the operating system dependent module converts the duplicated packet into an operating system compatible format prior to passing the duplicated packet to the operating system.
20. (Currently Amended) A method of transmitting a frame comprising:
obtaining a packet to be transmitted from an operating system;
converting the packet into an array based data structure that is independent of the operating system;
converting a virtual address of the array based data structure that reference one or more data buffers that store the packet into a physical addresses;
identifying logically contiguous data buffers associated with virtual addresses of the array based data structure that store the packet and converting the virtual address into a single physical addresses;
attaching the one or more data buffers to a transmit descriptor;
attaching optional information to the transmit descriptor by an operating system independent module;
transmitting the packet by a network device; and
freeing the one or more data buffers for other use after transmitting the packet.
21. (Original) The method of claim 20, wherein the packet is obtained from the operating system by an operating system dependent module in an operating system specific format.
22. (Currently Amended) The method of claim 20, further comprising coalescing the one or more data buffers into contiguous memory space dependent on a predetermined buffer size.

23. (Original) The method of claim 20, further comprising attaching the freed data buffers to a send queue for packets awaiting transmission.

24. (Canceled)

25. (Original) The method of claim 24, wherein the optional information includes a virtual local area network type.